

Chapter 11

Probabilistic Temporal Reasoning

[Steve Hanks & David Madigan]

Research in probabilistic temporal reasoning is devoted to building models of systems that change stochastically over time. Probabilistic dynamical systems have been studied in Statistics, Operations Research, and the Decision Sciences, though usually not with the emphasis on computational inference models and structured representations that characterizes much work in AI. At the same time, a related body of work in the AI literature has developed probabilistic extensions to the deterministic temporal reasoning representations and algorithms that have been studied actively in AI from the field's inception.

This chapter develops a unifying view of probabilistic temporal reasoning as it has been studied in the optimization, statistical, and AI literatures. It discusses two main bodies of work, which differ on their fundamental views of the problem:

- as a probabilistic extension to rule-based deterministic temporal reasoning models
- as a temporal extension to atemporal probabilistic models.

The chapter covers both representational and computational aspects of both approaches.

11.1 Introduction

Most systems worth modelling have some aspects of dynamics and some aspects of uncertainty. In many AI contexts, either or both of these aspects have been abstracted away, often because it was thought that probabilistic dynamic models were either impossible to elicit and construct, prohibitively expensive to use computationally, or both. Recent techniques for building structured representations for reasoning under uncertainty have made probabilistic reasoning more tractable, thus opening the door for effective probabilistic temporal reasoning.

This chapter surveys various systems, formal and computational, that have aspects of both uncertainty and dynamics. These systems tend to differ widely in how they define and attack the problem.

In providing a unified view of probabilistic temporal reasoning systems, we will address three main questions:

- *What is the formal model?* That is, how does the system represent system state, change, time, uncertainty? What kinds of change and uncertainty can the system express in principle, and how? What inference questions does the system address?
- *What is the representation?* Formal models can be implemented in many ways, and the representation for state, change, and uncertainty will affect the efficiency of inference.
- *What is the algorithm?* The formal model defines the inference task, and the representation specifies how the information is stored. How is the representation exploited to answer temporal queries?

11.2 Deterministic Temporal Reasoning

Temporal reasoning in the AI literature addresses the problem of inferring the *state* of a system at various points in time as it changes in response to *events*. This work has typically made strong certainty or complete-information assumptions, for example that the system's initial state is known, all events are known, the effects of events are deterministic and known, and any additional information provided about the system's state is complete and accurate.

Work in probabilistic temporal reasoning tries to relax some or all of these assumptions, addressing situations where the reasoner has partial information about the state and events, and where subsequent information can be incomplete and noisy.

We will begin with a summary of the deterministic problem, based on the Yale Shooting Problem example [Hanks and McDermott, 1987]. The problem consists of the following information, tracking the state of a single individual and a single gun

- The state is described fully by the propositions
 - A (the individual is alive)
 - L (the gun is loaded)
 - M (the gun has powder marks)
- The following events can potentially occur:
 - shoot: if the gun is loaded, this event makes A false, makes L false, and makes M true
 - load: if the gun is not loaded, makes L true, otherwise has no known effects
 - unload: if the gun is loaded, makes L false, otherwise has no known effects
 - wait: this event has no known effects

The effects of events are often described using logical axioms, which might take the following form for the events listed above:

$$\forall t. \text{true-at}(L, t) \wedge \text{occurs-at}(\text{shoot}, t) \Rightarrow \neg \text{true-at}(A, t + \epsilon) \wedge \neg \text{true-at}(L, t + \epsilon) \wedge \text{true-at}(M, t + \epsilon) \quad (11.1)$$

$$\forall t. \neg \text{true-at}(L, t) \wedge \text{occurs-at}(\text{load}, t) \Rightarrow \text{true-at}(L, t + \epsilon) \quad (11.2)$$

$$\forall t. \text{true-at}(L, t) \wedge \text{occurs-at}(\text{unload}, t) \Rightarrow \neg \text{true-at}(L, t + \epsilon) \quad (11.3)$$

where $t + \epsilon$ is the instant immediately following t^* .

One can pose inference problems of the following form: given (1) information about the occurrence of events at various points of time, and (2) direct information about the system's state at various point of time, infer the system's state at other points in time. The *prediction* or *projection* problem is the special case where the initial state and the nature and timing of events is known, and the system's state after the last event is of interest. In the *explanation* problem, information is provided about events and about the system's final state, and questions are asked about the system's initial state or more generally about earlier states. Both of these problems are special cases of the general problem of finding truth values for *all* state variables at *all* points in time, consistent with the constraints on event behavior—equations (11.1)–(11.3) above—and (partial) information about the system's state at *any* point in time.

This version of the temporal reasoning problem implicitly makes strong assumptions about the timing and duration of events, most notably that events occur instantaneously and affect the world immediately. In making these assumptions we ignore the large body of work on reasoning about durations, delays, and event timing summarized in [Schwalb and Vila, 1998]. We adopt this version

*The semantics of these logics typically model time points either as integers or as reals. The choice is unimportant for the analysis in this chapter. In the case of integer time points, $\epsilon \equiv 1$, and in both cases the notation $[t_i, t_j]$ refers to the closed interval between t_i and $t_j \geq t_i$

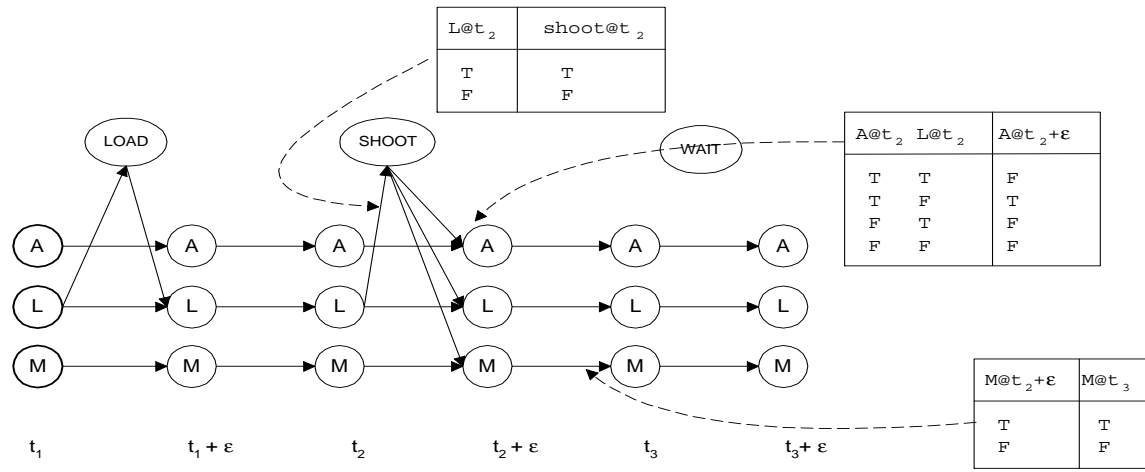


Figure 11.1: A structural graphical model for deterministic temporal reasoning

of the problem because it provides an easy bridge to the extant literature on probabilistic temporal reasoning, most of which makes these same assumptions. Some work has been done on reasoning with incomplete information about the timing and duration of events, which will be discussed below.

The original version of the Yale Shooting Problem is a projection problem:

- Initially (at t_1) A is true, and L is false. The initial state of M is not known.
- Load occurs at time t_1 , shoot occurs at $t_2 > (t_1 + \epsilon)$, and wait occurs at $t_3 > (t_2 + \epsilon)$
- The system's final state is to be predicted, particularly the state of A at some point $t_4 > (t_3 + \epsilon)$

A commonly studied explanation problem is to add the information that A was observed true at t_4 , and ask about the state of A or L at various intermediate time points. The technical difficulties associated with this problem are discussed in Section 11.6.1.

Graphical models Suppose it is known what events *occur* at what times. An event can occur but can *fail*, if its preconditions are not met. From this information and the event axioms (equations (11.1)–(11.3) above), we can build a *graphical model* representing the temporal scenario. The graphical model contains a node for each state variable at each relevant point in time—immediately before and immediately after the occurrence of each attempted event—along with a node representing the possibly successful occurrence of each event. Figure 11.1 shows the structure given only the information about event occurrences and the axiomatic information about their preconditions and effects.

Each node in this graph can be assigned a truth value. In the case of a proposition node, assigning a value of true simply means that the proposition was true at that time. In the case of an event node, a true value means that the event's precondition was true (the event occurred successfully).

In a deterministic setting, information or evidence takes the form of assigning a truth value to a node in the graph as is done with A and L in the initial state in Figure 11.2. At this point the temporal reasoning problem amounts to solving a constraint-satisfaction problem: given restrictions on truth-value assignments imposed by the evidence, by the event axioms, and by persistence assumptions (discussed below), find a consistent truth assignment for every node in the graph. Figure 11.2 shows the same structural model with partial information about the initial state and a consistent assignment of truth values to the nodes. The assignment need not be unique—in the example, the initial value of M was assigned arbitrarily.

Arcs in the graph represent dependencies among node values as suggested in the truth tables in Figure 11.1. These describe the effects of events, the effects of not acting, and other dependencies among state variables. There are three types of dependencies (constraints), discussed in turn.

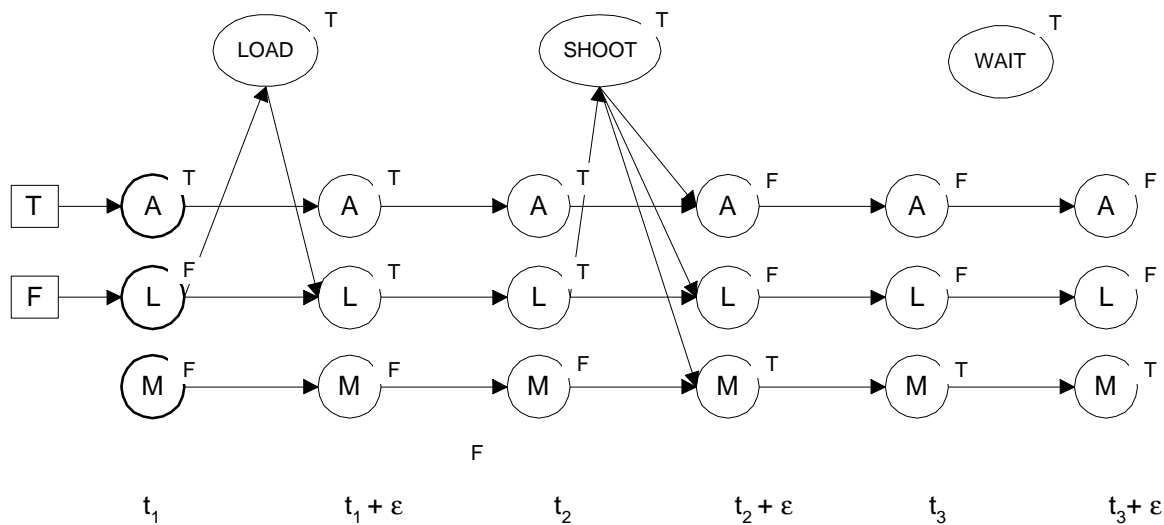


Figure 11.2: A deterministic model with evidence and inferred truth values

Causal constraints There are two sorts of causal constraints—the arrows linking events and propositions at proximate times—which describe an event’s preconditions and its effects. These are equivalent to the event axioms, Equations (11.1)–(11.3). For example, the dependencies linking *A* and *shoot* enforce the constraints described in Equation (11.1) describing the event’s immediate effects. The fact that there are only two arrows into the node representing $A@t_2 + \epsilon$ means that the variable’s value can be determined (only) from the previous state of *A*, $A@t_2$, along with information about whether *shoot* occurred successfully at t_2 . The truth table for this variable, pictured in Figure 11.1 reflects the implicit assumption that no event other than *shoot* occurs between t_2 and $t_2 + \epsilon$.

Persistence constraints The arcs from a proposition at one time point to the same proposition at the next time point were not mentioned explicitly in the problem description. These are called *persistence constraints*, and are equivalent to logical *frame axioms*. Persistence constraints enforce the common-sense notion that a proposition will change state *only* if an event causes it to do so. In the deterministic framework it is difficult to reason about events that *might* have occurred but were not *known* to occur, thus the assumption is made that the known events are the *only* events that occur, and thus no state variable changes truth value over an interval $[t_i + \epsilon, t_{i+1}]$, regardless of how much time elapses. Thus the truth tables for the persistence constraints always indicate that proposition *P* is true at t_{i+1} if and only if it was true at $t_i + \epsilon$.

There is a second implicit assumption in the diagram, which is that at a time point t_i , where an event is known to occur, the known event is the *only* event that occurs at that instant. Thus *A* will be false at $t_2 + \epsilon$ if and only if *shoot* was successful in making it false, or if it was already false. No event other than *shoot* can occur at t_2 to change *A*’s state.

There has been much research in the deterministic temporal reasoning literature on persistence constraints and the frame problem. This work, and its connection to probabilistic temporal reasoning, is discussed in Section 11.6.1.

Synchronic constraints Suppose that one observed over time that *L* was false whenever *M* was true. It might be convenient to note this observation explicitly in the graph, using an arc from *M* to *L* at every time point t . This is called a *synchronic* constraint, as it constrains the values of two state variables at the same point in time. The causal and persistence constraints are *diachronic* constraints, as they relate the values of state variables at different time points.

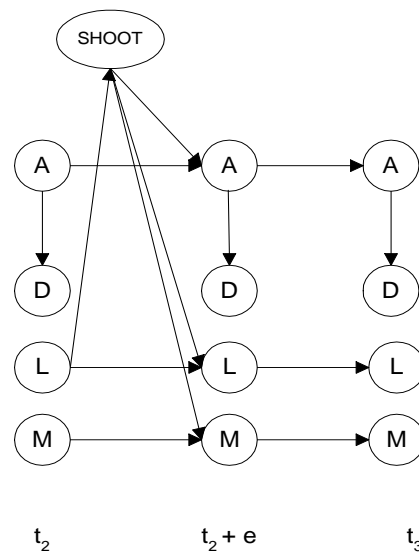


Figure 11.3: Syntactic synchronic constraints represent a definitional relationship between two propositions

Synchronic constraints are generally not formally necessary. For example, the relationship between M and L might be explained as follows:

1. initially M is always true
2. the only event that makes M true is shoot, which also makes L false
3. the only event that makes L true is load, but load never occurs after shoot occurs.

But all of these facts can be represented using diachronic constraints only—the synchronic constraints are redundant, though they might allow certain inferences to be made more efficiently. With redundancy also comes the possibility of contradiction: if an event were ever added that made M true without changing L, or if a load event ever occurred after a shoot, then the causal constraints would contradict the synchronic constraints.

Synchronic constraints are often used to represent simple syntactic *synonymy* or *antonymy* relationships: two propositions that *by definition* have the same or opposite states, and are included in the ontology simply for convenience. For example, we might introduce a state variable D, which is meant to be true if and only if A is false at the same time. This dependency can be enforced without explicit synchronic arcs in the graph, by ensuring that D and A are initially in opposite states, and that every action that makes A false makes D true, and vice versa. At best this method can be cumbersome, and subject to error. At worst it would be impossible to infer the relationship between A and D without the constraint, for example, if all that is known is that A is false at time t_4 .

It may therefore be more convenient to represent the synchronic constraint between A and D explicitly. In Figure 11.3, D is given a special status as an antonym for A: its state is determined *only* by the state of A at the same time point. Causal and persistence axioms are allowed to refer to A directly, but not to D, thus avoiding the potential inconsistency noted above. A is called the *primitive* variable and D is called the *derived* variable [Lifschitz, 1987].

In most deterministic temporal reasoning literature, synchronic constraints representing simple syntactic relationships are treated specially in this way, and event-induced synchronic constraints are not handled at all, since they add no expressive power to the model and are a possible source of inconsistency.* Synchronic constraints are more common in the probabilistic temporal reasoning literature, and are discussed again in Section 11.4.1.

* See [Ginsberg and Smith, 1988a] for an exception: a formal system that allows synchronic and diachronic constraints to be

Summary When the nature and order of events is known, a temporal reasoning problem can be represented as a graph where the nodes represent temporally scoped state variables and events. The arcs represent causal relationships (diachronic or synchronic) between the variables. The graph in Figure 11.1 was constructed from a set of axioms characterizing the domain, and has the following significant features:

- Causal relationships between variables caused by known events (the causal constraints) are all mediated through the event itself, and are not reflected in synchronic relationships among the state variables.
- Each state variable “persists” independently: whether or not a variable V changes state in the interval $[t_i, t_j]$ never depends on the state of another variable W .
- Events occur independently: the occurrence or non-occurrence of an event at one time does not affect whether subsequent events occur, though it may affect whether a subsequent event *succeeds*.

We now turn to various ways in which deterministic models for temporal reasoning can be given a probabilistic semantics which allows reasoning about incomplete information, stochastic events, and noisy observation information.

11.3 Models for Probabilistic Temporal Reasoning

We will consider several models for building probabilistic versions of these dynamic scenarios. We begin with models like the one above where events or actions are represented explicitly in the graph, and where the timing of the events is known. We begin by exploring the case where there is uncertainty as to *what* event occurs at a particular time. As a special case this allows reasoning about an event that might or might not occur.

In this section we will consider a simpler version of the example: the only state variables are A and L , the possible events are `load`, `shoot`, and `wait`, events occur at times t_1 and t_2 , and the temporal distance between t_1 and t_2 is known with certainty. Figure 11.4 shows the equivalent graphical model. It is identical in structure to the deterministic version, except for the additional event node E' (explained below), and the nature of the parameters noted on the figure analogous to the truth tables in Figure 11.1.

The main differences between this graph and the graph in Figure 11.1 are

- In Figure 11.4 there can be uncertainty as to *which* event occurred, so the event node is a random variable that ranges over all possible event types, whereas in Figure 11.1 the event type was fixed. In Figure 11.4 there are two nodes representing each event, a random variable representing which event occurred, and a second random variable representing that event’s effects.
- Nodes in the graphs are assigned probabilities rather than truth values.
- The constraints on the arcs represent probabilistic dependencies rather than deterministic dependencies.

We will introduce the following uncertainty in the model:

- Initially (at t_1), A is true with probability 0.9 and L is true with probability 0.5
- The `load` event makes L true with probability 0.8. It never causes L to become false, but with probability 0.2 it changes nothing.

mixed. They treat the case where blocking an air duct causes a room to become stuffy (a state variable), representing this as a synchronic constraint between `blocked` and `stuffy`.

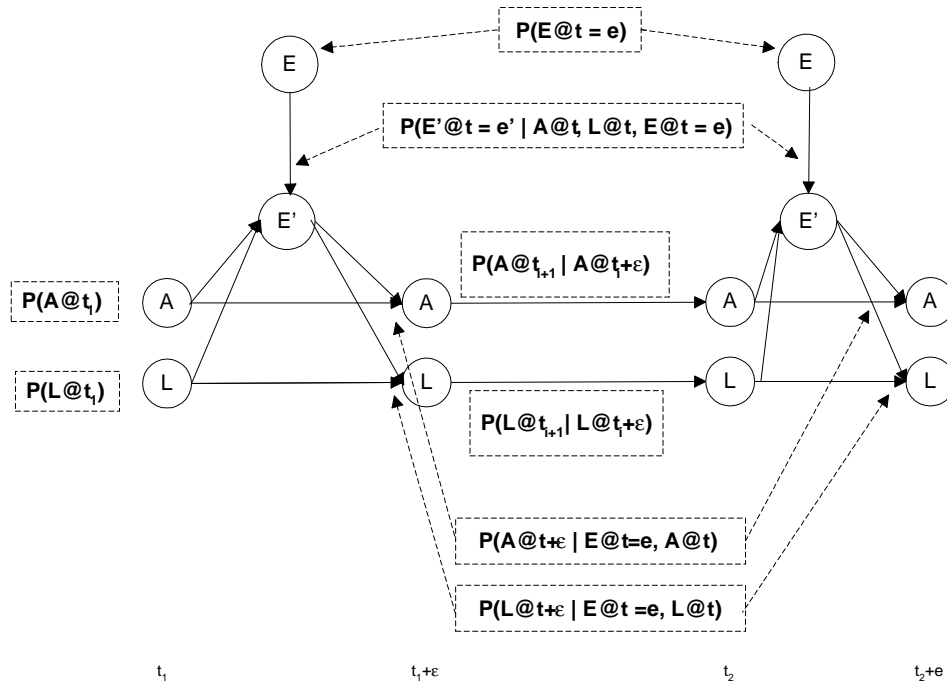


Figure 11.4: A probabilistic temporal model recording dependencies between events and states

- If L is true when `shoot` occurs, then with probability 0.75 A and L both become false, and with probability 0.25 L becomes false but A remains unchanged. If L is false when `shoot` occurs, then with probability 1 the event changes nothing.
- L can spontaneously become false, with probability .001, when `wait` occurs.
- Although it is known that events occur only at times t_1 and t_2 , there is uncertainty as to what event occurs at those times. At time t_1 , `load` occurs with probability 0.8 and `wait` occurs with probability 0.2. At time t_2 , `shoot` occurs with probability 0.8, `load` occurs with probability 0.1, and `wait` occurs with probability 0.1.

Let $P(A@t_i)$ be the probability that state variable A is true at time t_i given all available evidence and $P(E@t_i = e)$ be the probability that the event occurring at time t_i is the event e . The following model parameters are required:

- Probabilities describing the initial state of A and L : $P(A@t_1)$ and $P(L@t_1)$
- Probabilities describing which events occur: $P(E@t_i = e)$ for each i
- Probabilities describing the possible effects of an event that has occurred: $P(E'@t_1 = e' | E@t_i = e)$
- Probabilities describing the immediate effects of the events on the state variables: $P(A@t_i + \epsilon | E'@t_i = e', A@t_i)$ and $P(L@t_i + \epsilon | E'@t_i = e', L@t_i)$
- Probabilities describing what happens to the state variables during the time interval $[t_1 + \epsilon, t_2]$, an interval during which no event is known to occur: $P(A@t_{i+1} | A@t_i + \epsilon)$ and $P(L@t_{i+1} | L@t_i + \epsilon)$

11.3.1 Model structure

Each arc in the graph represents an explicit quantifiable probabilistic influence between the nodes it connects, for example that the value of $E'@t$ directly affects the value of $L@t + \epsilon$. The absence of arcs

in the graph implies certain probabilistic *independencies*. For example, information about the state of $A@t_1$ provides no additional information about the state of $L@t_1$. The variables $A@t_1 + \epsilon$ and $L@t_1 + \epsilon$ are probabilistically dependent, since the value of $E@t_1$ affects both, but become probabilistically independent if the value of $E@t_1$ is known*. It is again a significant feature of this model that there are no synchronic dependencies in the graph: all correlations between propositions at a single point in time, for example the relationship between $A@t_1 + \epsilon$ and $L@t_1 + \epsilon$, are caused by prior events.

Another significant feature of this model is that there is no way to represent dependencies over the occurrences of events, e.g. that `shoot` is more likely to occur at t_2 provided that `load` occurred at t_1 [†]. Section 11.4 will discuss the case where the distribution over event occurrences is state dependent.

11.3.2 Model parameters

In the previous section, the inference problem was defined as that of finding an assignment of truth values to every node in the graph, consistent with the explicit constraints. In the probabilistic case the problem is to construct a probability distribution over all nodes in the graph—the state of A and L at t_1 , $t_1 + \epsilon$, t_2 , and $t_2 + \epsilon$, and the value of E and E' at t_1 and t_2 , again consistent with the explicit probabilistic constraints and any available evidence. The question then arises: what probabilistic constraints are necessary to ensure a consistent and unique distribution exists?

Fundamental results from the general theory of probabilistic graphical models [Pearl, 1988] guarantee that the following parameters are necessary and sufficient to define a unique probability distribution over the nodes:

- Marginal (unconditional) probabilities for those nodes without parents: $\mathbf{P}(A@t_1)$ and $\mathbf{P}(L@t_2)$, and $\mathbf{P}(E@t_1 = e)$.
- A conditional probability table for each non-parent node conditioned on all possible values of its immediate parents. For example, the probability that A is true at $t_1 + \epsilon$ must be specified for all six combinations of the possible values for $E@t_1$ (`load`, `shoot`, `wait`) and the possible values of $A@t_1$ (`True`, `False`).

We discuss each class of parameters in turn.

Initial probabilities

Marginal probabilities for $\mathbf{P}(A@t_1)$ and $\mathbf{P}(L@t_1)$ are provided under the assumption that the values of these variables are probabilistically independent. Thus it is impossible to state that 85% of the time both A and L will initially be true, but 15% of the time they will both be false. This is due to our assumption that events cause all correlations. If such dependencies need to be represented, an “initial event” can be defined that induces the desired dependency.

Events and their effects

We reason about the effects of events in three stages:

- what event occurred
- what effects did the event have, given that it occurred
- what is the new state, given those effects

The first is determined by the marginal probability $\mathbf{P}(E@t_i = e)$. Note the assumption that this distribution is state independent. For the example, we have the following probabilities from the problem statement:

*These relationships depend on there being no evidence about temporally subsequent nodes in the graph. See [Cowell *et al.*, 1999], [Pearl, 1988] or [Charniak, 1991] for information about the exact set of independencies implied by this graph structure.

[†]It is the case, however, that information about what event occurred at t_1 along with information about what is true at $t_2 + \epsilon$ does affect the posterior distribution over the event that occurred at t_2 .

e	t_1	t_2
load	0.8	0.1
shoot	0.0	0.8
wait	0.2	0.1

The deterministic event representation was based on the idea of a *precondition*: if the event's precondition was true when it occurred, it was said to have *succeeded*, and it effected state changes. The effects of an event with a false precondition was not defined or the event was implicitly assumed to have no effects. In the present model, the concept of precondition and success is replaced by a more general notion of an event's effects depending on *context* (the prevailing state at the time of occurrence). There is no concept of a precondition: an event can occur under any circumstances, but its effects will depend on the context, and must be specified for all contexts.

Consider shoot for example, which was described above as having three possible outcomes depending on whether L is true. This event can be viewed as three “sub-events” shoot – 1, shoot – 2, and shoot – 3, each analogous to a deterministic event:

Event	Context	Probability	Effects
shoot – 1	L	0.75	-A, -L
shoot – 2	L	0.25	-L
shoot – 3	$\neg L$	1.00	

where +A means that the event causes A to be true regardless of its previous state, –A means that the event causes A to be false, and the absence of A in the effects list means that the event leaves A's state unchanged.

Thus the event shoot occurs exogenously, but there can still be uncertainty as to which of shoot-1, shoot-2, and shoot-3 occurs, and that uncertainty is context dependent. These are the probabilities $P(E'@t_i = e^i \mid E@t = e, S@t)$ where S is some subset of the state variables.

Once the nature of the sub-event is known, the resulting state $S@t + \epsilon$ is determined with certainty by the sub-event's list of effects. In other words, the quantity $P(S@t + \epsilon \mid E'@t = e', S@t)$ is deterministic, analogous to the truth tables in Figure 11.1. Therefore the state update is performed according to the following formula:

$$P(S@t_i + \epsilon \mid E'@t_i = e^i, S@t_i)P(E'@t_i = e^i \mid S@t_i, E@t_i = e)P(E@t_i = e)$$

As in the deterministic case it is assumed that ϵ is short enough that no other event occurs in the interval $[t, t + \epsilon]$, though probabilistic information about simultaneous events could easily be added to the model.

Alternative event models This event model (“probabilistic STRIPS operators” or PSOs) was introduced in [Hanks, 1990], and adopted in the design of the Buridan probabilistic planner [Kushmerick *et al.*, 1995]. It is well suited to situations in which events tend to change the state of several variables simultaneously, but suffers from the complexity of specifying events and sub-events, and the fact that the event probabilities are context dependent.

An alternative model works directly with context-independent events. The event probability measures only the probability that shoot occurs rather than wait or load, and does not measure the probability that shoot-1 occurs given shoot, for example. This moves the event's context dependence into the arcs governing how state variables change as a result of the event. Figure 11.5 shows two possible models. The leftmost is the PSO model described above, the second is a model that treats events as atomic and context independent. The additional complexity in the second model arises as a result of the fact that shoot tends either to change both A and L simultaneously, or to leave both unchanged. Thus load's state at $t + \epsilon$ depends on its prior state, whether or not shoot occurred, and whether A *changed state* from true to false (since if it did, load must have changed too). The synchronic arc from A to L is to allow reasoning about whether or not A changed state as a result of the event.

There are many different representations for events (see [Boutilier *et al.*, 1995b] for one alternative). Since most of them are formally equivalent [Littman, 1997], the choice of a particular model

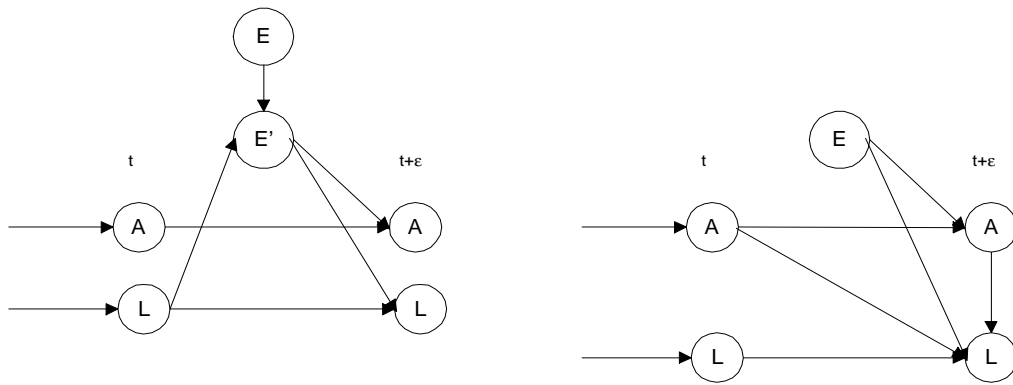


Figure 11.5: Alternative graphical models for representing the effects of events

would be made for reasons of parsimony or convenience of elicitation. See [Boutilier *et al.*, 1995a] for a more extensive comparison of event models.

“Persistence” probabilities

The last set of parameters describe the likelihood of state changes between the times events are known to occur. They are $P(P@t_{i+1} \mid P@t_i + \epsilon)$ and $P(P@t_{i+1} \mid \neg P@t_i + \epsilon)$, for each state variable P and each event time t_i . Again note that these dependencies are isolated to single propositions: knowing the state of L at t_i or whether L changes state between t_i and t_{i+1} does not affect the likelihood that A changes state in that interval of time. If there was a source of change known to change both simultaneously, it would have to be modeled as an explicit event that might or might not occur during the interval.

In the deterministic case these constraints were handled using either a monotonic or nonmonotonic closure axiom: the axiom(s) state that the known events are the only events, thus no proposition changes state between $t_i + \epsilon$ and t_{i+1} . In the probabilistic case the model accounts for the possibility that unknown events can occur during these intervals, thus there should be *some* likelihood that P changes state during $[t_i + \epsilon, t_{i+1}]$, and furthermore that probability will typically depend at least on the interval’s duration.

Persistence probabilities are typically specified using *survival functions*, which express the probability of a state-changing event occurring within an interval $[t, t + \delta]$ [Dean and Kanazawa, 1989]. These functions are often used as follows to express the persistence probabilities:

$$\begin{aligned} P(P@t + \delta \mid P@t) &= \epsilon^{-\alpha\delta} \\ P(P@t + \delta \mid \neg P@t) &= \epsilon^{-\beta\delta} \end{aligned}$$

where $\alpha, \beta > 0$, α measures the rate at which P will “spontaneously” become false and β measures the rate at which P will “spontaneously” become true.

One problem with using this functional form is that it confuses information about a *state change* with information about the proposition’s *new state*. That is, one might be certain that the proposition will change state at least once during an a long interval, but still might be unsure as to what its eventual state will be, as it might change several times.

In some cases the difference is unimportant: knowing that A changes state from true to false implies knowledge about its state at the end of the interval, since the probability of a state change back to true is 0.

In contrast, consider the problem of predicting whether or not a pet will be in a particular room. Over a long interval of time it is virtually certain that the pet will leave the room, but it might well return and leave several times over a long interval $[t, t + \delta]$, thus certainty about a state change does not amount to certainty about the new state, and the simple survivor function model will be inappropriate for reasoning about situations characterized by large values for δ .

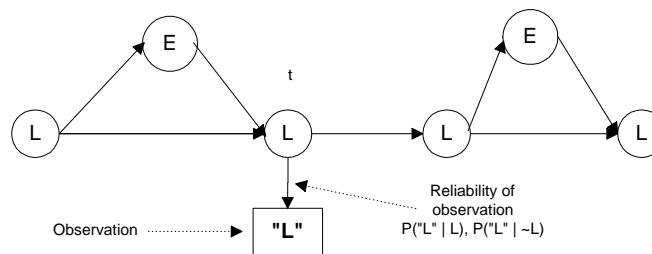


Figure 11.6: Adding evidence to the probabilistic graphical model

This particular form of the survivor function is still appropriate if δ is small enough that the probability of a *second* state change in the interval is improbable. In that case, information about the state change *is* equivalent to information about the new state. In the present model, however, the δ parameter represents temporal spacing between known events, and is not under our control, thus survivor functions of the above form might not be appropriate. Section 11.3.5 discusses two potential solutions to the problem: instantiating the model at more time points so the maximum δ makes the survivor model appropriate, and adopting a variant of the survivor model that explicitly differentiates between the probability of state changes and the proposition's state conditioned on the fact that it changed one or more times.

11.3.3 Evidence

We have not yet discussed the details of how to incorporate evidence about what facts are true or what events occurred. In the deterministic model, evidence took the form of knowing the value of various propositions at various points in time: the values of various nodes could be constrained to be true or false (see Figure 11.2). Evidence can likewise be placed on nodes in the probabilistic graphical model, with the added feature that information can be uncertain: the relationship between the evidence and the node's value is probabilistic rather than being limited to a deterministic setting of the node's value.

Figure 11.6 shows a case where evidence is received about the state of L at time t . As in Figure 11.2, an additional node is used to incorporate evidence into the graph, and the link between it and L 's actual state quantifies the relationship between the evidence and L 's actual state. In this model, the relationship between state and observation is state independent, though this assumption could easily be relaxed.

Two parameters are required to quantify this relationship $P('L'@t \mid L@t)$ and $P('L'@t \mid \neg L@t)$, where $'L'$ represents the *observation* that L was true at t , which might or might not reflect its true state at that time. These parameters reflect the probability that the evidence would have been observed assuming that L was true and false, respectively. The value of the $'L'$ node can be set to true—the fact that the observation *was made* is definitely true—and the propagation algorithms take care of the rest.

11.3.4 Inference

We have now discussed all parameters required to complete the model, and note that standard methods for probabilistic inference in graphical models [Pearl, 1988; Dawid, 1992] can be applied, which calculates probabilities for all variables and events at all points in time (i.e. for all nodes in the graph). These algorithms are “bi-directional” in that they consider the effect of forward causation (the effect of evidence on subsequent variables, mediated by the causal rules), and backward explanation (the effect of evidence on prior variables, again mediated by the rules). Using standard algorithms can be computationally expensive, however, and Section 11.5 discusses various methods for performing the inference efficiently.

11.3.5 Constructing the model

Most schemes for probabilistic temporal reasoning provide some method for constructing an appropriate network from model fragments representing the causal influences, event probabilities, and persistence probabilities. These pieces can be network fragments [Dean and Kanazawa, 1989], symbolic rules [Hanks and McDermott, 1994], or statements in a logic program database [Ngo *et al.*, 1995].

Since the model intersperses possible event occurrences with persistence intervals, the question arises as to which time points should appear explicitly in the graph. Not placing an event node at time t amounts to assigning a probability of zero to the occurrence of an event at that time, which could result in inaccurate predictions. On the other hand, the time required for the inference task grows exponentially with the number of nodes in the worst case [Cooper, 1990] is proportional to the size of the graph, so more nodes means costlier inference. This issue is particularly important when information about the occurrence of events is vague—if at *most* time points there is *some* probability that *some* event might occur.

A second consideration in constructing the graph was noted in Section 11.3.2: if survival functions are used for the persistence probabilities, and if there is the possibility of a proposition changing state more than once, then the interval between explicit events must be chosen so the probability of a second state change in the persistence interval is sufficiently small.

The most common approach to constructing the graph, [Dean and Kanazawa, 1989] for example, is to instantiate it on a fixed time grid. A fixed time duration Δt is chosen, and the model is instantiated at regular time points $t_1, t_1 + \Delta t, t_1 + 2\Delta t, \dots$ where t_1 is the first known time point: the time at which the first known event occurs, where the initial conditions are known, or the earliest time point at which temporal information is desired.

This approach is simple, and if Δt is chosen to be sufficiently small, will lead to an accurate predictive model. The problem with this approach is mainly computational: Δt must be chosen to satisfy the single-state-change assumptions for the *fastest-changing* state variable, and the model must be instantiated for all state variables at all time points, not just those temporally close to the occurrence of known events. This can lead to huge graphs containing long intervals of time where most or all of the state variables are extremely unlikely to change values. A projection or explanation algorithm must nonetheless compute probabilities for all events and all state variables at all time points.

In cases where there is a good model of when events occur, one might be able to instantiate event nodes only at times where events are likely to occur. The danger is that exponential survivor functions may be inappropriate given the longer interval between event instances.

An alternative model [Hanks and McDermott, 1994] instantiates the graph only at times when events are likely to occur, say $\{t_1, t_2, \dots, t_n\}$, which may be widely separated and irregularly spaced. Then two sets of persistence parameters are provided:

- The probability that a state variable P will undergo *at least one* state change in the interval $[t_i, t_{i+1}]$. This parameter depends only on $|t_{i+1} - t_i|$ (the time elapsed between t_i and t_{i+1}), and an exponential function is often appropriate.
- The probability that P will be true at t_{i+1} *provided* it changed state in the interval $[t_i, t_{i+1}]$.

This model has the advantage of parsimony, and also reflects a common-sense notion that many propositions have a “default” probability we can rely on when our explicit causal model breaks down. So the default probability for A is 0—if it changes state at all it will be to false, and will remain at false. On the other hand, the pet-prediction problem discussed in Section 11.3.2 is handled properly in that if the pet is assumed to move once, its position is predicted by the default probability, which is duration-independent.

Observation-based instantiation In some situations instantiation of the graph will be dictated by the environment itself. The model developed in [Nicholson and Brady, 1994] is an explicit-event model designed to monitor the location of moving objects. State variables store the objects’ predicted position and heading, and the events correspond to reports from the sensors that an object has moved from one region to another. Thus the events *indicate* rather than *initiate* change, and are observed

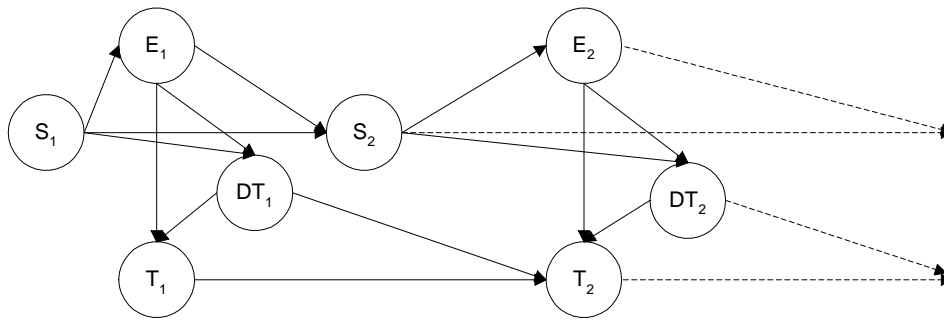


Figure 11.7: In a semi-Markov model the event times are also random variable

asynchronously. In the paper the assumption is made that the probability of a change in position over an interval is independent of the length of the interval, thus obviating the need for reasoning about unpredicted changes across irregularly spaced intervals. Work reported in [Goodwin *et al.*, 1994] is similar in that its events are actually observations of the state rather than change-producing occurrences. The work by Goodwin is oriented toward reasoning about how long propositions tend to persist, and does not involve a predictive model of how and when state variables might change state.

11.3.6 Summary

We have now developed a model for temporal reasoning that admits uncertainty about the initial state, about the effects of events, about the reliability of evidence, and about how the system changes due to unmodelled events that might occur over time. Inference methods are available to solve standard prediction and explanation problems.

We now discuss two relaxations to the model: cases in which there is a probabilistic model concerning the timing of events, and cases in which the system's state can influence the nature of subsequent events.

11.4 Probabilistic Event Timings and Endogenous Change

The work presented above assumed that although the exact nature of events was uncertain, their timing was known. A common relaxation of this model is to view the system as a *semi-Markov* process, in which the times at which events occur are also modeled as random variables. The models considered above were simple Markov processes: the system's current state is sufficient to predict (probabilistically) the system's next state, but the transition *time* is deterministic and instantaneous. A semi-Markov process assumes that both the nature of and the elapsed time to transition are unknown, but can be predicted probabilistically from the current state.

Semi-Markov processes are also amenable to graphical representations, though with increased complexity (Figure 11.7). S_i is the system's state when the i^{th} event occurs, E_i is the event that occurs, T_i is the time at which the i^{th} event occurs, and DT_i is the elapsed time between the i^{th} and $(i + 1)^{st}$ events. In this model (similar to one proposed in [Berzuini *et al.*, 1989]), both the time at which the i^{th} event occurred (T_i) and the transition time of the i^{th} event (DT_i) are represented explicitly, and the current state and the nature of the next event are sufficient to predict its duration.

An alternative temporal model proposed in [Berzuini *et al.*, 1989] and similarly in [Kanazawa, 1991] changes the interpretation of nodes in the graph. Instead of being random variables of the form $P@t$ ("P is true at t ") with range $\{\text{true}, \text{false}\}$, the nodes are taken to be the *times* at which events occur (random variables that range over the reals), so a node might then represent "the time at which P becomes true."

Instantaneous events are represented as a single node in the graph $\text{occur}(E)$, and facts (fluents) that hold over an interval of time are represented by instants representing when they begin and cease

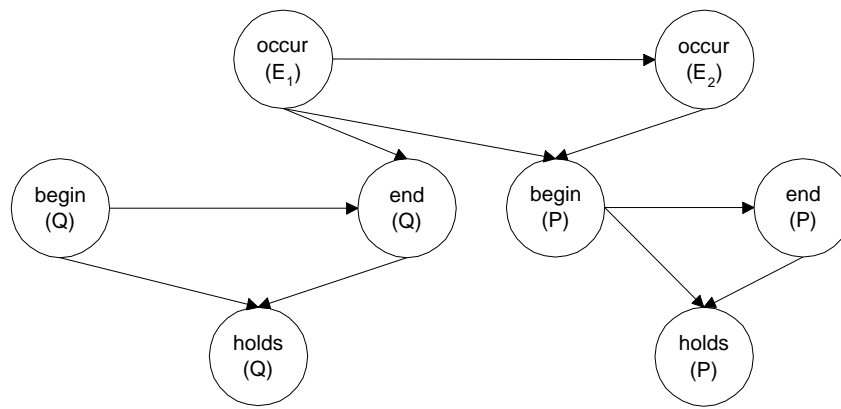


Figure 11.8: The “network of dates” model represents events and states implicitly but the time of occurrence explicitly

to be true along with a “range” node representing the interval of time over which they persist. Figure 11.8 shows an example where Q is known to be true at $t = 0$, event E_1 occurs making Q false and P true, followed by E_2 which makes P false.

This representation makes it easy to determine whether a particular variable is true at a point in time, but it can be expensive to discover whether *combinations* of variables are true simultaneously (as must commonly be done in establishing the context needed to predict an event’s effects). Also, neither Berzuini nor Kanazawa explain how the framework handles variables that change state several times over the course of a sequence of events, which is the central to the temporal reasoning problems commonly discussed in the literature.

The hidden Markov model framework has been successfully applied in contexts such as these. See, for example, [Ghahramani and Jordan, 1996] and [Smyth *et al.*, 1997]. There has also been recent work Bayesian analysis of hidden semi-Markov models [Scott, 2002].

Endogenous change

Berzuini addresses another problem, which is that the timing of one event can affect whether or not a subsequent event occurs. For example, a pump might or might not burn out (an event) depending on whether or not it first runs dry (another event), which in turn depends on whether a “refill” event occurs before the “runs dry” event. This sort of situation is not handled well by the models developed in Section 11.3, where the basic event probabilities are exogenous and state independent. Berzuini develops a theory whereby one event can *inhibit* the occurrence of a subsequent “potential” event, an event that might or might not occur. Non-occurrence is handled simply by letting its time of occurrence be infinitely large.

Event inhibition is just one aspect of a larger problem, which is that the system’s state can affect the occurrence, nature, and timing of subsequent events. This problem is generally called *endogenous change*, as the system’s state can endogenously cause changes whereas in the models discussed above, all change is effected by events that occur exogenously—they are specified externally and their occurrence is not affected by the system’s state.

The probabilistic model developed in this work can be extended to an endogenous-change model simply by allowing event-occurrence and persistence probabilities to depend on the state as well. The main problem is how to build and instantiate models of this sort: how and when should the model be instantiated to capture changes in state caused by endogenous events?

It is common to view the system’s endogenous change as being driven by a set of interacting *processes* which eventually will cause a state change [Barahona, 1994; Hanks *et al.*, 1995]. Taking an example from the latter source, consider a medical trauma case where the patient has suffered a blow to the head and to the abdominal cavity. These are both exogenous events, but they both initiate en-

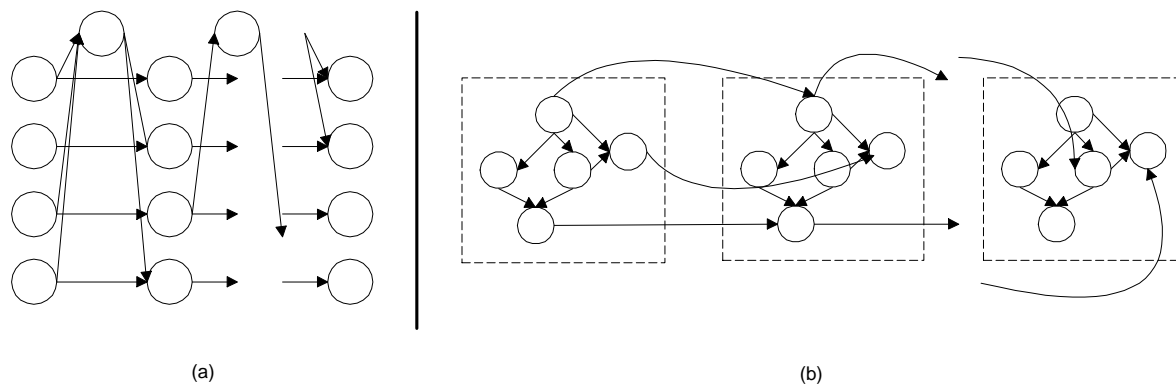


Figure 11.9: Explicit-event and implicit-event models have fundamentally different structure

ogenous change. The former causes the brain to begin to swell, which if left unchecked will lead to dilated pupils and eventually to loss of consciousness. The latter might cause internal bleeding, which will quickly cause a drop in blood pressure, light headedness, and eventually will also cause loss of consciousness. Administering fluids will tend to slow this process.

The next endogenous event might therefore be a change in state of the pupils, or the blood pressure, followed by another endogenous change if consciousness is lost. The fact that two forces lead to loss of consciousness might or might not make it occur sooner. And interventions (exogenous events) could change the nature of the change as well.

There are two main problems associated with reasoning about endogenous change: how to build the endogenous model, and how to make predictions efficiently. [Barahona, 1994] introduces model-building techniques based on ideas from qualitative physics, and a simulation technique called *interval constraining*. In [Hanks *et al.*, 1995] a system is presented where the endogenous model is built by aggregating sub-models for the various forces acting on the system. The inference technique, based on *sequential imputation*, is discussed in Section 11.5. [Aliferis and Cooper, 1996] develop a formalism called Modifiable Temporal Belief Networks that allows expressing endogenous causal mechanisms through an extension to standard temporal Bayesian networks; they do not discuss inference algorithms.

11.4.1 Implicit event models

The models considered to this point have assumed that the source of change in the system, the modeled events, could be predicted or observed, and their effects on the system assessed accurately. This is consistent with the deterministic temporal reasoning literature, and appropriate for most planning and control applications.

In contrast, consider a case where observable exogenous interventions are rare, but one is allowed to observe all or part of the system state at various points in time. Medical scenarios are good examples, since exogenous events (interventions) are rare relative to the significant unobserved endogenous events that occur. In this case the explicit-event model may not be adequate to reason about the system, since so little information about the occurrence or effects of events is available.

An implicit-event model also depicts the system at various points in time, but there are no intervening causal events to provide the structure for predicting change. One primary difference between explicit- and implicit-event models is the role played by synchronic constraints (probabilistic dependencies among variables at a single point in time). While these dependencies are ubiquitous in real systems, it was unnecessary to represent them explicitly in the explicit-event models developed above, since it was reasonable to assume that all synchronic dependencies were caused by the modelled events. In implicit-event models, the absence of events means that observed synchronic dependencies must be noted explicitly in the model.

Figure 11.9 compares an abstract explicit-event model (a) with an implicit-event model (b). In the implicit-event case we see a sequence of static (synchronic) probabilistic models representing the system state at points of observation, connected by some number of diachronic constraints. Two main questions thus arise:

- What should the synchronic model look like at various points in time, and in particular should the synchronic model be the *same* at every time point?
- What diachronic constraints should be added to connect the static models, and in particular should the pattern of diachronic connections be the same at every time point?

The work presented in [Provan, 1993] is an example of how implicit-event models are built. The paper presents a dynamic model for diagnosing acute abdominal pain, which is based primarily on a static model constructed by a domain expert. The static model is duplicated at various time slices, presumably including those in which observations about the patient's state are made. There is no procedure presented for determining which diachronic arcs should be included in the model. The paper points out that models of this sort can be too big to support efficient inference, and presents several techniques for reducing the model's size. As such, it answers neither of the questions posed above.

Another example of an implicit-event model is presented in [Dagum and Galper, 1993], designed to predict sleep-apnea episodes. The input in this work is a sequence of 34,000 data points representing a patient's state measured at closely spaced regular time intervals. Each data point consists of four readings: heart rate, chest volume, blood oxygen concentration, and sleep state. The problem is to predict the onset of sleep apnea before it occurs.

This problem is an interesting contrast to the explicit-event models studied above, in that no explicit information about events is available and the state information is insufficient to build an effective process model, but large amounts of observational data are available.

In this case a k -stage temporal model—both synchronic and diachronic components—is learned from the observational data*, where k is a user-supplied parameter.

The value of state variable X_i at time t is then predicted by combining the value predicted by the diachronic model with the value predicted by the synchronic model. If $\pi(X_{it})$ is the set of all synchronic dependencies involving X_i and $\theta(X_{it})$ is the set of all diachronic dependencies involving X_i , then the value of X_{it} is computed according to the formula:

$$\mathbf{P}(X_{it} \mid \pi(X_{it}), \theta(X_{it})) = (1 - \alpha_{it})\mathbf{P}(X_{it} \mid \pi(X_{it})) + \alpha_{it}\mathbf{P}(X_{it} \mid \theta(X_{it}))$$

where α_{it} determines how strongly the new prediction depends on prior information mediated by the diachronic model as opposed to current information mediated by the synchronic model. Although α is time dependent, the paper does not mention how it might vary over time.

Summary At this point in time there is a stark contrast between temporal reasoning work based on explicit-event versus implicit-event models. The former is mainly concerned with building probabilistic models from more primitive components (rules, model fragments, logical axioms) that represent a causal or functional model of the system. The key issues here are what form the primitives take, and how they are pieced together to produce an accurate and efficient predictive model of the domain. In contrast, the implicit-event work has been oriented more toward providing special-purpose solutions to particular problems, and toward developing techniques to aid a human analyst in constructing these special-purpose models from data. There is less emphasis on causal or process models, and on automated model construction. In the current literature on implicit-event models, there is no generally satisfactory answer to the two questions posed at the beginning of this section—what should the synchronic model look like, and what diachronic constraints are appropriate—particularly regarding how the diachronic part of the model is built.

*The paper also alludes to “refining the model with knowledge of cardiovascular and respiratory physiology, during the process of model fitting and diagnostic checking,” but does not explain this refinement process.

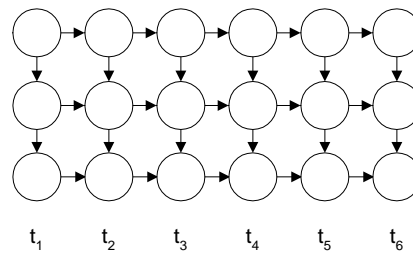


Figure 11.10: A simple dynamic belief network on a fixed time grid

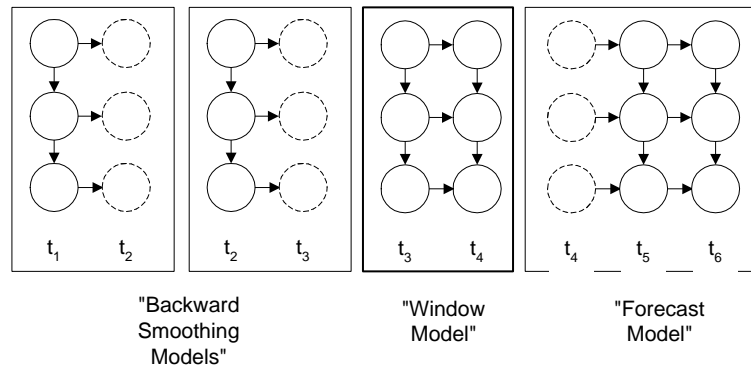


Figure 11.11: The Simple Dynamic Belief Network Decomposed

11.5 Inference Methods for Probabilistic Temporal Models

As we mentioned in Section 11.3.4, standard algorithms for probabilistic inference in graphical models apply directly to the kinds of models we have been discussing—see, for example, [Jensen, 2001], [Pearl, 1988], [Cowell *et al.*, 1999], or [Dawid, 1992]. However, as modeling progresses temporally, inference becomes increasingly intractable.

11.5.1 Adaptations to standard propagation algorithms

A number of authors have described variants on the standard algorithms that take advantage of the temporal nature of the models—key references include [Kjaerulff, 1994], [Provan, 1993], and [Dagum and Galper, 1993]. Though these references differ somewhat in their specific implementations, the essential idea is to maintain a model “window” containing a modest number of time slices. Computations in this window are carried out using standard algorithms; as time progresses, the window moves forward, relying on the Markov properties of the model—the past is conditionally independent of the future given the present—to maintain inferential veracity. This windowing idea enables standard algorithms to be applied to infinitely large models.

Here we sketch the elements of Kjaerulff’s algorithm using a simple example. Figure 11.10 shows a stochastic temporal model with six time slices labeled one to six. Kjaerulff’s algorithm decomposes the basic model into zero or more backward smoothing models each focusing on a single time slice, a window model containing one or more time slices, and a forecast model containing zero or more time slices. Figure 11.11 shows a decomposition for our simple example. Note that the forecast model contains not only time slices five and six, but also the vertices from time slice four required to render slices five and six conditionally independent of the remainder of the model. Similarly, the backward smoothing models contain the vertices required to render them conditionally independent of future models. The algorithm ensures that the window model has absorbed all evidence from previous time

slices; inference within the window then uses standard algorithms to further condition on evidence pertaining to the time slices within the window. “Backward smoothing” is a process whereby evidence is passed backwards from the window to the previous time slices using a message passing approach. “Forecasting” is carried out using a Monte Carlo algorithm.

Perhaps the most challenging aspect of Kjaerulff’s algorithm involves moving the window. This he accomplishes by first expanding the model and the window, and then reducing the window and dispatching some time slices from the window to the backward smoothing model. Thus, window expansion by, say, k new time slices consists of (a) adding k new consecutive time slices to the forecast model, (b) moving the k oldest time slices of the forecast model to the time window, and (c) “compiling”^{*} the newly expanded window. Window reduction involves elimination of vertices from the window and an updating of the remaining probability to reflect evidence from the eliminated variables—see [Kjaerulff, 1994] for details. We note that there are close connections between Kjaerulff’s algorithm and the forwards-backwards algorithm used in Hidden Markov Modeling [Smyth *et al.*, 1997].

Unfortunately, the computations involved in window expansion and reduction, as well as the computations required within the window can quickly become intractable. Several authors have proposed approximate inference algorithms - see, for example, [Boyen and Koller, 1998] or [Ghahramani and Jordan, 1996]. Recently the stochastic simulation approach has attracted considerable attention and we discuss this next.

11.5.2 Stochastic simulation

Stochastic simulation methods[†] for temporal models provide considerable flexibility and apply to very general classes of dynamic models. The state-of-the-art has progressed rapidly in recent years and we refer the reader to [Doucet *et al.*, 2001] for a comprehensive treatment. In what follows, we draw heavily on [Liu and Chen, 1998]. [Kanazawa *et al.*, 1995] also provide an overview but less general in scope. We note that while our focus in this Chapter is on probabilistic inference for stochastic temporal models, the methods described here also apply to statistical learning for temporal models, as well as applications such as protein structures simulation, genetics, and combinatorial optimization. We start with a general definition:

Definition 11.5.1. *A sequence of evolving probability distributions $\pi_t(\mathbf{x}_t)$, indexed by discrete time $t = 0, 1, 2, \dots$, is called a probabilistic dynamic system.*

The state variable \mathbf{x}_t can evolve in several ways but generally in what we consider \mathbf{x}_t will increase in dimension over time, i.e., $\mathbf{x}_{t+1} = (\mathbf{x}_t, x_{t+1})$, where x_{t+1} can be a multidimensional component. [Liu and Chen, 1998] describe three generic tasks in systems such as these: (a) prediction: $\pi_t(x_{t+1} \mid \mathbf{x}_t)$; (b) updating: $\pi_{t+1}(\mathbf{x}_t)$ (i.e., updating previous states given new information); and (c) new estimation: $\pi_{t+1}(x_{t+1})$ (i.e., what we can say about x_{t+1} in the light of new information)?

The models described in this Chapter fit into this general framework. More specifically they are *State Space Models*. Such models comprise two parts: (1) the observation equation, which can be formulated as $y_t \sim f_t(\cdot \mid x_t, \phi)$; and (2) the state equation, $x_t \sim q_y(\cdot \mid x_{t-1}, \theta)$. The y_t are observations and the x_t are the observed or unobserved states. Of interest at any time t is the posterior distribution of $\mathbf{x}_t \equiv (\phi, \theta, x_1, \dots, x_t)$. Hence the target distribution at time t is:

$$\pi_t(\mathbf{x}_t) = p(\phi, \theta, x_1, \dots, x_t \mid \mathbf{y}_t) \propto p(\theta, \phi) \prod_{s=1}^t f_s(y_s \mid x_s, \phi) q_s(x_s \mid x_{s-1}, \theta).$$

These models arise in, for example, signal processing, speech recognition, multi-target tracking problems, computer vision, DNA sequence analysis, and financial stochastic volatility models.

Simple Monte Carlo methods for dynamic systems such as these require, for each time t , random samples drawn from $\pi_t(\mathbf{x}_t)$. Many applications require more general schemes such as importance

^{*}The standard Lauritzen-Spiegelhalter algorithm involves “moralization” and triangulation of the DAG to create an undirected hypergraph in which computations take place. This process (which is NP-hard) is often called compilation.

[†]Also known as Monte Carlo methods.

sampling. Even then, most published methods assume that all of the random draws obtained at time t are discarded when the system evolves from π_t to π_{t+1} . *Sequential* Monte Carlo methods, on the other hand, “re-use” the samples obtained at time t to help construct random samples at time $t + 1$, and offer considerable computational efficiencies. The basic idea dates back at least to [Hendry and Richard, 1990]. See also [Kong *et al.*, 1994] and [Berzuini *et al.*, 1997]. Here we reproduce the general formulation of [Liu and Chen, 1998]. We begin with a definition:

Definition 11.5.2. Definition 2 A set of random draws and weights $(x^{(j)}, w^{(j)})$, $j = 1, 2, \dots$ is said to be properly weighted with respect to π if:

$$\lim_{m \rightarrow \infty} \frac{\sum_{j=1}^m h(x^{(j)}) w^{(j)}}{\sum_{j=1}^m w^{(j)}} = E_{\pi}(h(X))$$

for any integrable function h .

The basic idea here is that we can come up with very general schemes for sampling x_t ’s and associated weights, so long as the weighted average of these x ’s is the same as the average of x ’s drawn from the correct distribution (i.e., π). In particular, we do not have to draw the x_t ’s from π_t , but instead can draw them from a more convenient distribution, say g_t . Liu and Wong’s Sequential Importance Sampling (SIS) proceeds as follows:

Let $S_t = \{\mathbf{x}_t^{(j)}, j = 1, \dots, m\}$ denote a set of random draws that are properly weighted by the set of weights $W_t = \{w_t^{(j)}, j = 1, \dots, m\}$ with respect to π_t . Let H_{t+1} denote the sample space of X_{t+1} , and let g_{t+1} be a trial distribution. Then the SIS procedure consists of recursive applications of the following SIS steps. For $j = 1, \dots, m$,

(A) Draw $X_{t+1} = x_{t+1}^{(j)}$ from $g_{t+1}(x_{t+1} | \mathbf{x}_t^{(j)})$; attach it to $\mathbf{x}_t^{(j)}$ to form $\mathbf{x}_{t+1} = (\mathbf{x}_t^{(j)}, x_{t+1}^{(j)})$.

(B) Compute

$$u_{t+1}^{(j)} = \frac{\pi_{t+1}(\mathbf{x}_{t+1}^{(j)})}{\pi_t(\mathbf{x}_t^{(j)}) g_{t+1}(x_{t+1}^{(j)} | \mathbf{x}_t^{(j)})}$$

and let $w_{t+1}^{(j)} = u_{t+1}^{(j)} w_t^{(j)}$.

It is easy to show that $(\mathbf{x}_{t+1}^{(j)}, w_{t+1}^{(j)})$ is a properly weighted sample of π_{t+1} .

For State Space models with known (ϕ, θ) , Liu and Chen suggest the following trial distribution:

$$g_{t+1}(x_{t+1} | \mathbf{x}_t) \propto f_{t+1}(y_{t+1} | x_{t+1}, \phi) q_{t+1}(x_{t+1} | x_t, \theta)$$

with

$$u_{t+1} = \int f_{t+1}(y_{t+1} | x_{t+1}, \phi) q_{t+1}(x_{t+1} | x_t, \theta) dx_{t+1}.$$

Hanks *et. al.* [Hanks *et al.*, 1995] describe a particular implementation of this scheme, called *sequential imputation*. Other choices of g are possible - see, for example, [Berzuini *et al.*, 1997]. [Liu and Chen, 1998] describe various elaborations of the basic scheme including re-sampling steps and *Local SIS* and go on to describe a generic Monte Carlo algorithm for probabilistic dynamic system. Recent work on these so-called “particle filters” by Gilks and Berzuini [Gilks and Berzuini, 2001] is especially ingenious.

In summary, stochastic simulation methods apply to very general classes of models and extend to both learning algorithms as well as probabilistic inference. This flexibility does come at a computational cost however; while SIS is considerably more efficient than non-sequential Monte Carlo methods, the ability of the algorithm to scale to, for example, thousands of variables, remains unclear.

11.5.3 Incremental model construction

The techniques discussed above were based on the implicit assumption that a (graphical) model was constructed in full prior to solution. Furthermore, the algorithms computed a probability value for every node in the graph, thus providing information about the state of every system variable at every point in time. For many applications this information is not necessary: all that is needed is the value of a few *query variables* that are relevant to some prediction or decision-making situation. Work on incremental model construction starts with a compositional representation of the system in the form of rules, model fragments, or other knowledge base, and computes the value of a query expression trying to instantiate only those parts of the network necessary to compute the query probability accurately. In [Ngo *et al.*, 1995], the underlying system representation takes the form of sentences in a temporal probabilistic logic, and constructs a Bayesian network for a particular query. The resulting network, which should include only those parts of the network relevant to the query, can be solved by standard methods or any of the special-purpose algorithms discussed above.

In [Hanks and McDermott, 1994] the underlying system representation consists of STRIPS-like rules with a probabilistic component (Section 11.3.2). The system takes as input a query formula along with a *probability threshold*. The algorithm does not compute the exact probability of the query formula; rather it answers whether or not that probability is less than, greater than, or equal to, the threshold. The justification for this approach is that in decision-making or planning situations, the exact value of the query variables is usually unimportant—all that matters is what side of the threshold the probability lies. For example, a decision rule for planning an outing might be to schedule the trip only if the probability of rain is below 20%.

The algorithm in [Hanks and McDermott, 1994] works as follows: suppose the query formula is a single state variable $P@t$, and the input threshold is τ . The algorithm computes an estimate of $P@t$ based on its current set of evidence. (Initially the evidence set is empty, and estimate is the prior for $P@t$). The estimate is compared to the threshold, and the algorithm computes an answer to the question “what evidence would cause the current estimate of $P@t$ to change with respect to τ ?”

Evidence and rules can be irrelevant for a number of reasons. First, they can be of the wrong sort (positive evidence about P and rules that make P true are both irrelevant if the current estimate is already greater than τ). A rule or piece of evidence can also be too tenuous to be interesting, either because it is temporally too remote from the query time point, or because its “noise” factor is too large. In either case, the evidence or rule can be ignored if its effect on the current estimate is weak enough that even if it were considered, it would not change the current estimate from greater than τ to less than τ , or vice versa.

Once the relevant evidence has been characterized, a search through the temporal database is initiated. If the search yields no evidence, and the current qualitative estimate is returned. If new evidence is found, the estimate is updated and the process is repeated.

There is an aspect of dynamic model construction in [Nicholson and Brady, 1994] as well, though this work differs from the first two in that it constructs the network in response to incoming observation data rather than in response to queries.

For work on learning dynamic probabilistic model structure from training data, see, for example, [Friedman *et al.*, 1998], and the references therein.

11.6 The Frame, Qualification, and Ramification Problems

No survey of temporal reasoning would be complete without considering the classic frame, qualification, and ramification problems. These problems, generally studied in the deterministic arena, have been central to temporal reasoning research since the problem was first discussed in the AI literature. Does a probabilistic model provide any leverage in solving these problems?

11.6.1 The frame problem

The frame problem [McCarthy and Hayes, 1969; Shanahan, 1987] refers to the need to represent the “common-sense law of inertia,” that a variable does not change state unless compelled to do so, say by the occurrence of a causally relevant event. In the shooting scenario discussed in this chapter, common sense says that the L proposition should not change as a result of the W event occurring, even though there may be no axioms explicitly stating which state variables W does *not* change.

There is a practical and an epistemological aspect to the problem. As a practical matter, in most theories, most events leave most variables unchanged. Therefore it is unnecessarily inconvenient and expensive to have to state these facts explicitly. And even if the tedium could be engineered away, the user may lack the insight and detailed information about the domain necessary to build a deterministic model—one where every change and non-change is accounted for properly and explicitly. A complete and correct event model may be impossible.

Probabilistic theories in themselves do not constitute a solution to the practical problem of enumerating frame axioms, but neither do they stand in the way of a solution. Just as deterministic STRIPS operators embody the assumption that all variables not mentioned should remain unchanged, structured probabilistic action representations like the probabilistic STRIPS operators discussed in Section 11.3.2 can do the same. The practical side of the frame problem is addressed by choosing appropriately structured representations, irrespective of the model’s underlying semantics. See [Boutilier and Goldszmidt, 1996] for an extensive analysis of the role of structured action representation in ameliorating the problem of specifying frame axioms.

The epistemological problem acknowledges the fact that information about events and their effects will typically be incomplete. As a result, inferences can be incorrect and might be contradicted by subsequent information that exposes gaps in the reasoner’s knowledge. In terms of the frame problem this means that persistence inferences (e.g. that A persists across a W event or over a period of time where no event is known to occur) should be defeasible: they might need to be retracted if contradicted by subsequent evidence (an observation that A was in fact false).

A probabilistic model confronts this problem directly. First, it provides an explicit representation for incomplete information about events and their effects, and separates what is known about the domain (information about event occurrences and their effects) from what is not known (the probabilistic components of the event description, and the probabilistic persistence assumptions). Second, it requires *quantifying* the extent to which the model is believed complete: noise terms in the event descriptions measure confidence in the ability to predict their effects, event and persistence probabilities measure confidence in the ability to predict the occurrence of events and the extent to which modeled events are sufficient to explain all changes.

It is instructive to point out why the Yale Shooting Problem does not arise in the probabilistic model. The problem originally arose in attempting to solve the frame problem using one defeasible rule: prefer scenarios that minimize the number of “unexplained” changes. The problem was that there were two scenarios minimal in that regard, one (intuitive) scenario in which $load$ made L true, $shoot$ made A false, and $wait$ left A false, and another (unintuitive) scenario in which $load$ made L true, L spontaneously became false shortly thereafter, and $shoot$ left A true. Since both scenarios involved two state changes, the nonmonotonic logic frameworks were unable to identify the intuitive scenario as preferable to the unintuitive one.

Both scenarios are possible under the probabilistic framework, but there is an *explicit* model parameter measuring the likelihood of L spontaneously changing from true to false, which can be considered relative to the likelihood that $shoot$ causes a state change. If this change is (relatively) unlikely, then the intuitive scenario will be assigned a higher probability. Thus the problem is solved at the expense of having to be explicit and numeric about one’s beliefs.

11.6.2 The qualification problem

The qualification problem [Shoham and McDermott, 1988; Ginsberg and Smith, 1988b] involves the practical and epistemological difficulty of verifying the preconditions of events. The most common example involves a rule predicting that turning the key to the car will cause the car to start, *provided*

there is fuel, spark, oxygen available, no obstruction in the tailpipe, and so on, *ad infinitum*. The practical problem is that verifying all these preconditions can be expensive; the epistemological problem is that enumerating necessary and sufficient conditions for an event's having a particular effect will generally be impossible.

The epistemological part of the qualification problem amounts to admitting that the *stated* necessary and sufficient conditions might be incomplete. Once again, this problem can be addressed deterministically by allowing the event axioms to be defeasible [Shoham, 1988]: “if all of an event's *stated* preconditions are met, then *defeasibly* conclude that the event will have its predicted effects.” In other words, there is some possibility that there is some unknown precondition that will prevent the event from having its predicted effects.

The probabilistic model addresses this possibility in that it requires an explicit numeric account of the likelihood that an event will have its effects, conditioned on the fact that its context (precondition) holds in the world. That is, the event specification describes the likelihood that an effect will not be realized even though the context holds, and also the likelihood that an effect will be realized even though the context does not hold.

Although the probabilistic framework does not itself address the “practical” qualification problem (the computational difficulty of verifying the known context), it allows computational schemes that do address the problem. Suppose that the inference task specified *how certain* a decision maker must be that an event produce a particular effect. In that case, it might be possible to avoid verifying every contextual variable, because one could demonstrate that the effect was *sufficiently certain* even if a particular precondition turned out to be false. This mode of reasoning, which is enabled because the probabilistic framework allows the notion of *sufficiently certain* to be captured explicitly, is discussed in Section 11.5.3 and in more detail in [Hanks and McDermott, 1994].

11.6.3 The ramification problem

The ramification problem concerns reasoning about an event's “indirect effects.” An example from [Ginsberg and Smith, 1988a] is that moving an object on top of a ventilation duct has the immediate effect of obstructing the duct, and in addition has the secondary effect of making the room stuffy. They express this relationship as a synchronic rule of the form “obstructed duct implies stuffy room” which is true at all time points. The technical question is whether formal temporal reasoning frameworks, particularly those that solve the frame and qualification problems nonmonotonically, handle the synchronic constraint properly. For example, if the inference that the vent was blocked was arrived at defeasibly, and if subsequent evidence reveals that the duct was in fact clear, will the (defeasible) inference that the room is stuffy be retracted as well?

As we have seen, probabilistic temporal reasoning systems have not addressed the interplay between synchronic and diachronic constraints in any meaningful way, and generally a probabilistic model will use one but not the other.

On the other hand, the example above could more properly be handled in a framework that treats the stuffiness as an endogenous change in the model rather than as a synchronic invariant. In that case work on endogenous change models (Section 11.4) would be relevant, though the probabilistic semantics sheds no additional light on the problem.

In summary, these classic problems have both epistemological and computational aspects. Probabilistic models address the epistemological issues directly in that they require the modeler to quantify his confidence in the model's coverage of the domain, a concept that can be difficult to capture in a satisfying manner with a nonmonotonic logics.

Probabilistic models can exacerbate the computational problems worse in that there are simply more parameters to assess. On the other hand, a numeric model admits approximation algorithms and other techniques for providing “accurate enough” answers, which could make inference easier (Section 11.5.3).

11.7 Concluding Remarks

We have presented a variety of approaches to building and computing with models of probabilistic dynamical systems. Most of this work adopts one of the following sets of assumptions:

- (Explicit-event models) A good predictive model of the domain is available and the important causal events are observable or controlled. As a result the events can be included explicitly in the model, the predictive model determines the diachronic dependencies, and synchronic dependencies are rare. The emphasis is on eliciting realistic causal models of the domain, and building the model on demand from smaller fragments.
- (Implicit-event models) Observational data about the system's state are plentiful, though one cannot count on observing or predicting the causally relevant events, and in many cases a compelling causal model will not be available. The absence of explicit events means that both synchronic and diachronic dependencies are important, and the challenge is determining the network's structure. This is typically viewed as a learning task, and success is measured by how well the model fits the available data rather than whether the model is physically plausible.

The main challenges facing the field at this point involve

- more expressive models
- automated model construction
- integrating explicit- and implicit-event models
- scaling to larger problems

First, the models studied in this chapter have been propositional. Although it is unlikely that efficient general-purpose algorithms will emerge for systems as powerful as first-order probabilistic temporal logics [Haddawy, 1994], computing with models that allow limited quantification seems possible.

Second, several automated model construction techniques were studied in the chapter, but most either assumed known exogenous events, or adopted the time-grid approach to building the model which is likely to be infeasible for large models instantiated over long periods of time. Building parsimonious models on demand, especially in situations where endogenous change is common, is a key challenge for making the technology widely useful.

Third, we noted the disparity between explicit- and implicit-event approaches. Clearly no situation will fit either approach perfectly, and a synthesis will again produce more widely applicable systems.

Finally, realistic system models may have thousands of state variables evaluated over long intervals of time. The need to make inferences from these models in reasonable time poses severe challenges for current and future probabilistic reasoning algorithms.